GraVol

(Graph-Volume)

Travis Jones

Final Report on Week 15

External Consultants: Dr. Julie Hruby and Dr. Martin Veillette

Date Submitted: 12/12/2011

Table of Contents

1. Application Development Section

Project Concept Proposal

Purpose: This program is to be able to read in an image of a drawing of a standard type used by archaeologists to depict the profile of ceramics. Once it is read in, the program will calculate the volume that vessel would be able to hold.

- **Context**: Museums today have a large quantity of ancient pottery, much of it broken. These vessels each have drawings made up of the entire profile (inside and out). It is impractical to piece many of them back together as most museums are lacking in storage space. Besides which, it could possibly damage them to attempt many methods of determining their volume. It is also impractical to try to calculate the volume by hand if being based on the drawings.
- **Goals**: The program will make it easy and practical to calculate the volume of ancient and, likely, broken ceramic vessels.
- **Audience**: This program will primarily be geared towards archaeologists, art historians, and museum workers, some of which would like the chance to be able to study ancient measurement systems.
  #Updated 9/18/11
- **Functionality**: By the completion of this project, the user will be able to browse for a scanned image (an uncompressed image format such as BMP and TIF) of the drawing they intend to process, enter in the maximum diameter or height, determine the content level, and then calculate the volume of the selected area of the vessel. It will also have a fully functioning GUI. #Updated 12/12/2011
- **Milieu**: There was one similar product called Vase (© Greg Christiana, 1994) which is now obsolete. It only worked up to Mac OS 9. It also calculated the volume based on the outer profile of the vessel, giving the misleading calculation of including the vessel itself in the volume. It would also require that the user draw on the computer screen with a mouse or tablet to determine the profile. All of these rendered the program both inaccurate, and now incompatible on current operating systems.
  #Updated 9/11/11
- **Novelty**: This program will be written in Python, an operating system independent language, so that it will retain its functionality regardless of the computer. It will also derive its calculations based on the inner profile of the drawing, and thus provide better accuracy. It will be computer based, and not web based, so that it is accessible even when internet is not.

Resources:

- Python Programming Language: Python 2.7 #Updated 9/10/11
  - http://www.python.org/
- Python Imaging Library (PIL) #Updated 9/10/11
  - www.pythonware.com/products/pil/
- TKinter
- Pyinstaller 1.5.1 #Added 12/10/11
  - www.pyinstaller.org/
- Images from already drawn out pottery with known measurements.
- DIA Diagram Editor

    - http://dia-installer.de/ #Added 9/25/11
- Camtasia 7.1 #Modified 12/10/11
- CamStudio 2.0 #Added 11/20/11
- Xvidcap 1.1.7 #Added 12/11/11
    - http://sourceforge.net/projects/xvidcap/
- **Archaeological Resource(s)**:
    - Article with reference to the program "Vase" (© Greg Christiana, 1994) http://kuscholarworks.ku.edu/dspace/bitstream/1808/5495/1/Younger_Metron.pdf

Challenges:

- Learning Python.
- Learning to handle images, and process them in a way to distinguish between the lines normally used on graph paper and the drawn lines of the vessel.
- Developing a GUI.

Measures: There will be several criteria that must be met: intuitive operation of the program, easily navigable design, accurate volume calculation.

- Operation and design: Once several people who are not very technically inclined are able to operate the program comfortably after having read the instructions, the program will be considered usable.
- Accurate Measurement: Once the program is able to successfully calculate the volume of several different styles of pottery, while giving a reasonably small and explicit margin of error, it will be considered reliable. Modified 12/10/2011

Updated 9/25/11

Future Extensions:

- The ability to compensate for a vessel being illustrated with handles.
- Image editing tools.
- Weight calculations for the vessel itself.
- Redesign of the GUI
- Accuracy ranges.
- Calculation of a vessel volume limit in addition to the already implemented ability to calculate the volume to a certain point in the vessel.
- Minimum volume capacities for a vessel based on a sherd.

Inspiration:

- **Motivation**: I was first told about the need for this kind of software during my first semester of college by archaeologist and art history professor Dr. Julie Hruby. Back then, I was still only an art history major. Now that I've got computer science as a second major, it would make sense to do a project that could benefit those of my other major who sorely lack programmers.
- **Profession**: This project will help me get experience with working on a more long term project than I am used to and will accustom me to more timeline oriented programming. It will also allow me to gain experience in more diverse programming problems, such as in dealing with images, and learning new programming languages.
- **Other (Optional):** To add to the interest of this program, by learning the different measurement systems of different locations throughout history it will help determine whether

the original location of the pottery was governed by the same ruler. This is possible, considering that it would be unlikely that two different towns would have been ruled by the same person but not have the same systems of measurement.

Test Plans:

Test-GV01

- **Features to be tested**
  - Getting the interior of a shape.
  - Getting an outline of a shape.
  - Bridging gaps in a shape.
  - Straighten an image according to the nearest edge.
- **Approach**
  - Test each individual function with shapes that will hopefully represent the most difficult case possible when in use.
- **Suspension criteria and resumption requirements**
  - If any particular test on any image can cause a critical error, the test will stop, find the source of the problem and fix it.
- **Environmental Needs**
  - Dell Latitude E5400
  - Operating Systems: Ubuntu 11.10, Windows 7
  - Software Requirements: PIL (Python Imaging Library), Python, Tkinter
- **Schedule**
  - Date to successfully calculate volume for ideal image: 11/6/2011
  - Date to successfully calculate the volume of a drawn image: 11/13/2011
- **Acceptance criteria**
  - Accurately calculate the volume of a vessel, whether by hand or digitally to within 5% of reality.
  - Intuitive use of the program.
- **Roles and responsibilities**
  - Test each function of the program under the most extreme case imaginable for the application, both drawn by hand, and digitally.

Test-GV02

- **Features to be tested**
  - Calculate the volume of an object.
- **Approach**
  - Test the program on ideal images of ideal (square) shapes.
- **Suspension criteria and resumption requirements**
  - If any particular test on any image can cause a critical error, the test will stop, find the source of the problem and fix it.
  - If there is any inaccuracy in the volume calculation the test will be paused until the error is found.
- **Environmental Needs**
  - Dell Latitude E5400
  - Operating Systems: Ubuntu 11.10, Windows 7
  - Software Requirements: PIL (Python Imaging Library), Python, Tkinter
- **Schedule**

    - Date to successfully calculate volume for ideal image: 11/13/2011
- **Acceptance criteria**
    - Accurately calculate the volume of an ideal vessel to 100% accuracy.
- **Roles and responsibilities**
    - Test the accuracy of the volume calculations against objects with known volumes.

Test-GV03

- **Features to be tested**
    - Test the GUI through all its features.
- **Approach**
    - Test each individual function of the GUI, both forwards and backwards, and present it with unusual situations in an attempt to break it.
- **Suspension criteria and resumption requirements**
    - If any particular test on any image can cause a critical error, the test will stop, find the source of the problem and fix it.
- **Environmental Needs**
    - Dell Latitude E5400
    - Operating Systems: Ubuntu 11.10, Windows 7
    - Software Requirements: PIL (Python Imaging Library), Python, Tkinter
- **Schedule**
    - Successfully run the program through the GUI with no errors, while trying to break it: 11/20/2011
- **Acceptance criteria**
    - Avoid any critical errors that the user may cause.
    - Intuitive use of the program.
- **Roles and responsibilities**
    - Test each function of the program under the most extreme case imaginable for the application, both drawn by hand, and digitally.

Test-GV04

- **Features to be tested**
    - GUI
    - Volume Calculation
- **Approach**
    - Test the program in its entirety through the GUI.
    - Compare the volume calculation of an irregular shape with its known volume.
- **Suspension criteria and resumption requirements**
    - If any particular test on any image can cause a critical error, the test will stop, find the source of the problem and fix it.
- **Environmental Needs**
    - Dell Latitude E5400
    - Operating Systems: Ubuntu 11.10, Windows 7
    - Software Requirements: PIL (Python Imaging Library), Python, Tkinter
- **Schedule**

    - Date to successfully calculate the volume of a drawn image: 11/13/2011
    - Successfully run the program through the GUI with no errors: 11/20/2011
- **Acceptance criteria**
    - Be unable to incur any critical error.
    - Calculate the volume within 5% of accuracy on trial vessel.
    - Intuitive use of the GUI.
- **Roles and responsibilities**
    - Test all available features under the most challenging conditions.
    - Recruit several people to try out the program from start to finish.

Software Demos:

- **GraVol v1.0 Demo Video**
    - Note: Please turn on captions for the step-by-step instructions.
    - Link: http://www.youtube.com/watch?v=pC8VReY83q4

Vision and Scope: #Modified 12/10/11

GraVol will be able to reveal many different aspects of long buried and shattered pottery, including the weight and volume of the contents it once held, the volume of the vessel itself, based on only a scanned drawing of the vessel and its dimensions. This program will also be able to use images of a variety of file types that are the most popular among archaeologists.

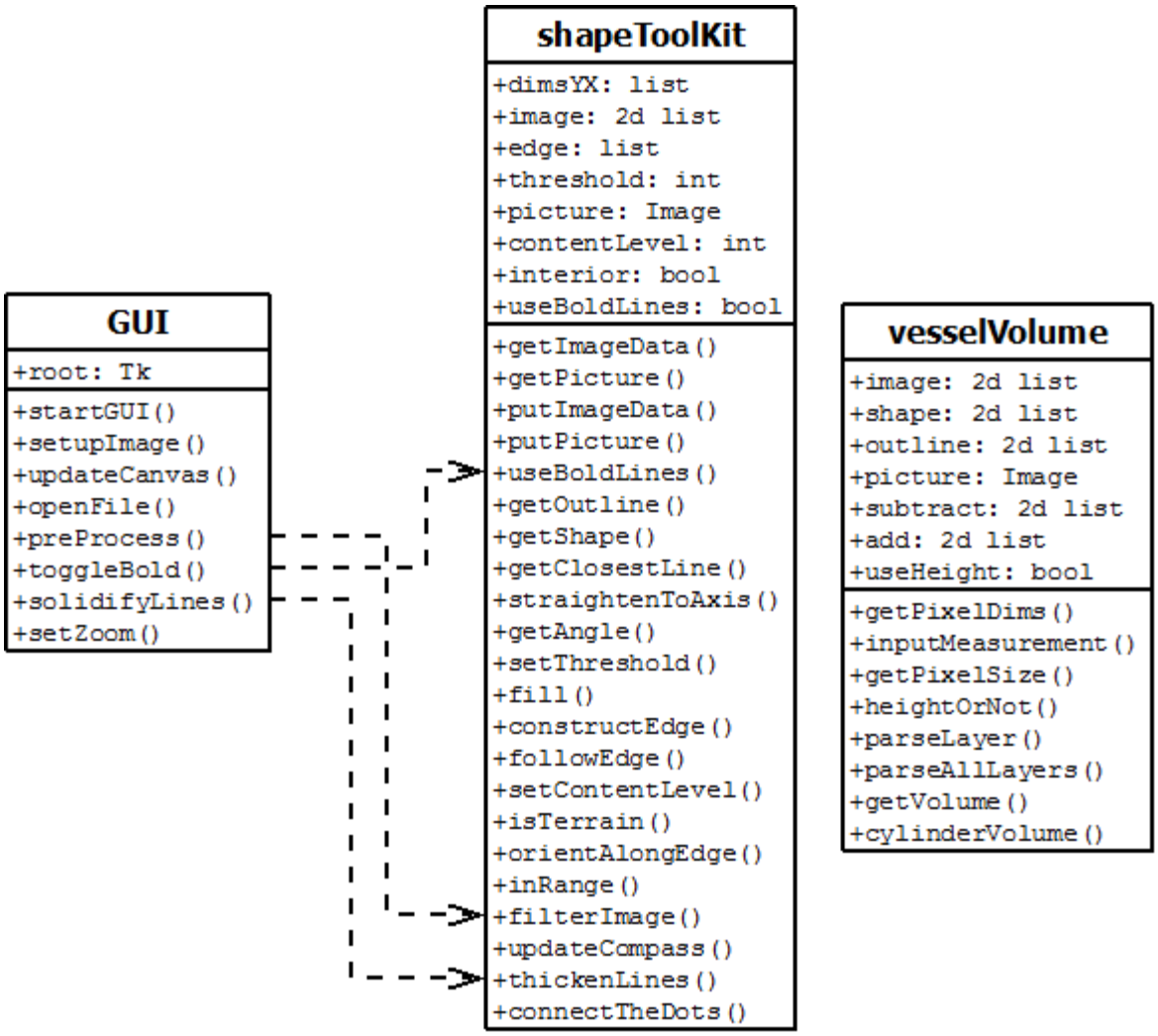
Unfortunately, due to the limited amount of time available during the project period, certain elements of the program will be omitted until time allows for further work on the project. Therefore, what will be immediately essential to the launch of the program will be that it be able to calculate the volume of the vessel, it will still be able to use all of the popular image formats. (Note: Further details on the current goals of the program are available in the Purpose portion of this document, above.)

Software Requirements Specifications #Updated 9/18/11

1. Read an image file.
    - Evaluation Method: Entire image contained in memory, and all metadata and individual pixels easily accessible and manipulable.
    - Dependency: None
    - Priority: High
    - Requirement revision history: 9/18/11- No longer focused on RAW file format. The image library already accepts multiple file types, which removed that requirement.
2. Display an image.
    - Evaluation Method: Successfully display an image scaled to fit inside the computer screen.
    - Dependency: 1 (reading an image).
    - Priority: High
    - Requirement revision history: None
3. Fill an amorphous area on a drawing with color.
    - Evaluation Method: Completed once all pixels within certain colored boundaries are filled with color, regardless of the area's shape.
    - Dependency: 1 (reading an image); 2 (displaying an image).
    - Priority: High
    - Requirement revision history: None
4. Create an image window that is manipulable.
    - Evaluation Method: Once there is a window that is able to accept some minor edits and is embedded within an interface it will be considered successful.
    - Dependency: 1 (reading an image); 2 (displaying an image).
    - Priority: Medium
    - Requirement revision history: 9/18/11- Took place of GUI requirement.
5. Implement water level and depth markers.
    - Evaluation Method: Requires an intuitive way to set the water level of the vessel, and a marker to distinguish the distance from the base to the interior depth.
    - Dependency: 1 (reading an image); 2 (displaying an image).
    - Priority: Medium
    - Requirement revision history: None

6. Create a function to allow selection of the area to be calculated for volume. #Edited 10/16/2011
    - Evaluation Method: Will be complete once the user is able to click inside the enclosed space.
    - Dependency: 1 (reading an image); 2 (displaying an image); 5 (water level marker- this will limit the selected area from including space outside the vessel).
    - Priority: Medium
    - Requirement revision history: 9/18/11- Took place of GUI requirement.
7. Implement a volume calculation function.
    - Evaluation Method: Create a function that will be able to calculate the volume based on the number of pixels and the image resolution.
    - Dependency: None
    - Priority: Low
    - Requirement revision history: None.
8. Design GUI layout. #Updated 9/25/11
    a. Evaluation Method: Once there are three people who agree that it is both intuitive and esthetically pleasing, and it having all the necessary tools incorporated will it be complete. (Note: The decision to have three people agree on the aesthetic and intuitive properties of the GUI layout was purely arbitrary.)
    b. Dependency: None
    c. Priority: Medium
    d. Requirement revision history: 9/18/11- Took place of GUI requirement.
9. Get outline of object. #Added 10/16/2011
    a. Evaluation Method: Get outline of any shaped object.
    b. Dependency: 1 (reading an image); 2 (displaying an image).
    c. Priority: Medium
    d. Requirement revision history: None
10. Parse shape to calculate volume. #Added 10/16/2011
    a. Evaluation Method: Select appropriate pixels from a row of them and select them for either addition or subtraction from the volume.
    b. Dependency: 1 (reading an image); 2 (displaying an image); 4 (fill an amorphous space).
    c. Priority: Medium
    d. Requirement revision history: None

2. System Design and Architecture



Made using Dia Diagram Editor- (http://dia-installer.de/)

3. Known Bugs and Issues- GraVol v1.0

#Updated 12/10/2011

This version of GraVol is fully functional. It is able to calculate the volume of vessels based on their drawings to a very high degree of accuracy.

- On the Windows 7 operating system, as it is running the process of hardening the edges the display freezes. It does not actually impact the program. The program resumes once the process has finished.
- Only BMP and TIFF files are currently supported, and only in RGB mode.
- .EXE files are only available for Windows 7 and Ubuntu 11.10, on 32-bit systems. For all other systems, the source code must be used, and Python, Tkinter, PIL must be installed.
- If a drawing is being used that is not very well delineated, then it will need to be edited using a program such as MS Paint or Adobe Photoshop to fill in edges which are not dark enough, or remove interfering shapes such as smudges.
- There are instances when excess noise in the drawing will cause GraVol to not fill in a selected Volume Area with blue. It is unclear yet how this affects the volume calculations. But, though this did happen during a few test cases, a very high degree of accuracy was maintained.
- There must be a break in the top of the drawing, along the rim. GraVol is dependent on exploring the exterior of the drawing, which it cannot do if there is not a way to exit the volume area.
- GraVol has not yet been tested using illustrations made on mechanical design programs such as AutoCAD, which would give the most accurate test results. Most current tests were done on hand-made illustrations, making the accuracy of the volume calculation dependent on the accuracy of the illustration.

4. About the Author

Travis Jones- Travis has a very diverse set of interests, just a few of which include both the making and history of art, and also in computer programming. He is in his final year at Berea College, where he will graduate with a double major in Art History and Computer Science. He hopes to soon either go on to graduate school, or get a career in software engineering.

5. Executive Section

[Insert Logo Here]

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 8/4/2011

Accomplishments: I got in contact with Dr. Julie Hruby, archaeologist and art history professor at Berea College, to arrange a meeting, and also to learn if there were any standards that were used when making drawings of ancient ceramics. I have begun looking at some resources for learning Python. I spent some time coming up with a project name. I spent some time reviewing some old emails and information on the project that I put together over the summer in preparation for this project.
Challenges: Coming up with a project name that was simple enough to be reasonable for software took a bit of effort.
Time Spent: 3 hours on the proposal, 15 minutes on the Executive section, 1 hour reviewing information accumulated during summer.
Goals: Meet with my Project Director. Make a few simple programs in Python to gain some experience with it. Meet with Dr. Julie Hruby to discuss drawing standards.

[Insert Logo Here]

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 9/12/2011

Accomplishments: I made some simple programs, and have some of the fundamental concepts down on how to use Python. I discussed my project with my Project Director and my class and got some ideas on how to work on the project. I downloaded PIL, an imaging library for Python. I decided to make the program with Python 2.7 instead of 3.2, as there are more libraries created for it. I made some simple functions in Python for calculating volume. I decided to work on reading in a RAW image file first instead of a BMP since it is the same on any operating system.
Challenges: I had to fumble through compiling my first Python program. I solved some problems with installing PIL.
Time Spent: 45 minutes working on the volume formulas. 1 hour studying PIL and solving some of the errors in compiling. 3 hours working on this week's report, and planning out how to proceed.

Goals: I plan to read in a simple image, display it, and manipulate it. I will begin research on suitable GUI software for program. Begin work on the function to fill all of an amorphous space.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 9/18/2011

Accomplishments: I read in both a TIF and a BMP with no trouble. PIL is apparently able to process several different file types with the same function. I was also able to easily output the values of the individual pixels and the image resolution. I drew a preliminary logo for the program. I revised my project requirements.
Challenges: I had to spend a lot of time coming up with and drawing a suitable logo. It took a bit of trial and error to figure out the workings of Python and converting lists into a string, as I am used to C++.
Time Spent: 2 hours finding out how to use images with the PIL library. 8 hours developing a logo. 2 hours revising the SRS portion of this document. 3 hours working on my resume.
Total time: 24 hours.
Goals: Fill an amorphous space with color.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 9/25/2011

Accomplishments: I was able to fill the amorphous space on a text-based map using my new class, Probe. All it needs is a couple minor changes to work with the image, and to be able to accommodate thresholds. I was able to make a UML to represent as much as I can at this stage how the software architecture will look.
Challenges: Learning the UML software, Dia Diagram Editor.
Time Spent: 8 hours on the Probe class; 3 hours on the UML.
Total time: 33 hours.
Goals: Apply the Probe class to an image, and beginning research on building a GUI, and what it entails.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 10/10/2011

Accomplishments: I completely remade my function to fill an amorphous space on a bitmap, this time in an incredibly more efficient way, and with far higher accuracy.
Challenges: Perfectly filling a space, and presenting challenges to it that could break the program and overcoming these breaks.
Time Spent: 12 hours on the fillSpace class (includes making the README.txt file); 30 minutes on Executive section of this document, 30 minutes updating the UML.
Total time: 46 hours.
Goals: Implement the volume finding functions, and begin research on building a GUI, and what it entails.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 10/17/2011

Accomplishments: I updated some errors in the fillSpace function. I restructured several functions, implemented a function to return an outline of the nearest shape. I also renamed to shapeFunctions to encompass several functions, and heavily restructured it. I also implemented a new class called vesselVolume a function to parse a shape and select certain lengths for adding volume and subtracting volume, where appropriate.
Challenges: Overcoming some quirks/glitches in the Python language. Trying to find an appropriate architecture for the classes. Debugging the fillSpace function.
Time Spent: 5 hours debugging the fillSpace function (now named “fill”). 6 hours building the vesselVolume class to the point that it can parse the rows of pixels in a shape. 4 hours restructuring the shapeFunctions class and adding several functions to make it more object oriented. 1 hour updating this report.
Total time: 62 hours.
Goals: Finish calculating the volume of a shape with a known volume. Create a function to distinguish between the exterior height and the interior depth of a shape.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 10/23/2011

Accomplishments: I implemented a terminal-based way for the user to test the program. I also worked on a way to increase the contrast, and patch a shape with broken lines. As of yet, the patching doesn't work. I also practiced some on GUIs.

Challenges: Some odd glitches that kept on occurring, without any real explanation for why they happen. They randomly started, then randomly went away. It may have simply been a syntax error, but requires further investigation.

Time Spent: 3 hours working on learning to program GUIs. 15 hours spent trying to find a way to overcome containers with broken lines. 30 minutes updating the report and the README file. 45 minutes creating a terminal-based interface.

Total time: 81 hours 15 minutes.

Goals: Debug my current problems, and learn how to patch a line. Create a basic GUI for the program.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 10/30/2011

Accomplishments: Made the capability of travelling around an object which has holes in it. Made a small, and simple GUI (still no functionality).

Challenges: Bridging gaps in a drawing in a diagonal direction. Learning the basics of GUIs. Battling with thresholding problems.

Time Spent: 3 hours learning and programming a GUI. 12 hours creating a search pattern for the nearest pixel that is the continuation of a shape.

Total time: 81 hours 15 minutes.

Goals: Create a fully functional GUI. Refine my thresholding. Fix my volume calculation function.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 11/8/2011

Accomplishments: Completely refined shape detection functions. I produced a semi-functional GUI. I improved the volume calculator to within 99.25% accuracy on ideal shapes.
Challenges: Restructuring my edge detection techniques. Applying new GUI methods.
Time Spent: 10 hours producing a semi-functional way to bridge gaps in a shape. 1 hour discarding previous method of bridging gaps and producing a fully functional alternate method. 6 hours producing a GUI. 3 hours testing the volume calculator.
Total time: 116 hours 15 minutes.
Goals: Complete the GUI. Improve the accuracy of volume calculation for ideal shapes to 100%.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 11/13/2011

Accomplishments: I have produced a fully functioning GUI, complete with zoom, instructions, and fairly intuitive use.
Challenges: Learning about events, and bindings. Learning to apply zoom, and compensating for click coordinates when zoomed in and out, and scrolled.
Time Spent: 6 hours completing the GUI. 3 hours applying the zoom. 1 hour asking random people to run my program with minimal input from myself. 1 hour attempting to test the program with ideal shapes.
Total time: 126 hours 15 minutes.
Goals: Add in the functions to flip and rotate the image. Acquire drawings of vessels with known volumes. Test the program on real ceramics with known volumes.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 11/20/2011

Accomplishments: I have added in the abilities to flip and rotate the image. I have also added in a status bar for the long processes. I also debugged some issues on returning to prior progress.
Challenges: Learning how to apply a progress bar with accurate progress representation.
Time Spent: 3 hours implementing a status bar. 3 hours adding the image rotating and mirroring. 30 minutes updating this file. 3 hours creating the demo video. 30 minutes testing the program on Windows.
Total time: 136 hours 15 minutes.
Goals: Test the program on Mac computers. Acquire drawings with known volumes to test against.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 11/28/2011

Accomplishments: I have added more detailed instructions, along with the ability to flip between different instructions. I have also added in a timer to disable the fill function after a few seconds, in the event that it enters an infinite loop due to noise.
Challenges: None.
Time Spent: 2 hours implementing the timer. 2 upgrading the instructions options.
Total time: 140 hours 15 minutes.
Goals: Test the program on Mac computers. Acquire drawings with known volumes to test against. Test the ability of the program to work with many different drawings.

To: Dr. Jan Pearce, Project Director
From: Travis Jones
Subject: GraVol
Date: 12/10/2011

Accomplishments: I made executable files for both Windows 7 and Ubuntu 11.10. I made minor modifications to the GUI. I designed my poster for the Poster Presentation. I edited several details of my logo, including adding in a small watermark. I remade my demo video.

Challenges: Debugging Pyinstaller, which I used to create the executable files. I tried to incorporate my logo into the window bar, with no success.

Time Spent: 4 hours designing my poster. 6 hours creating executable files. 4 hours remaking the demo video. 3 hours trying to incorporate my logo into the window design. 1 hour updating my report.

Total time: 158 hours 15 minutes.

Goals: N/A